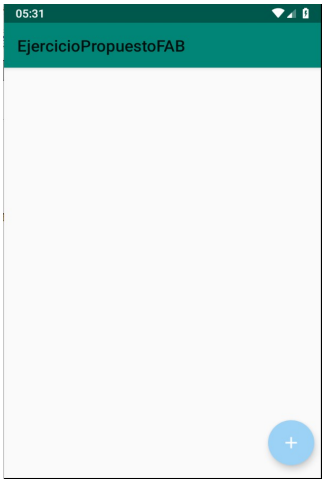
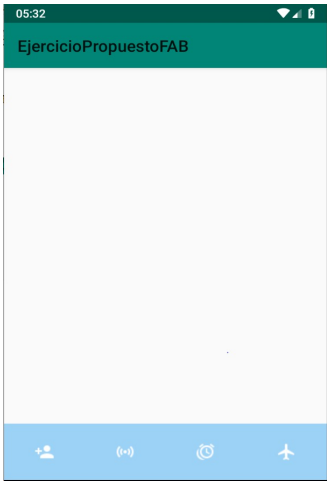
Transformar Floating ActionButton en ToolBar

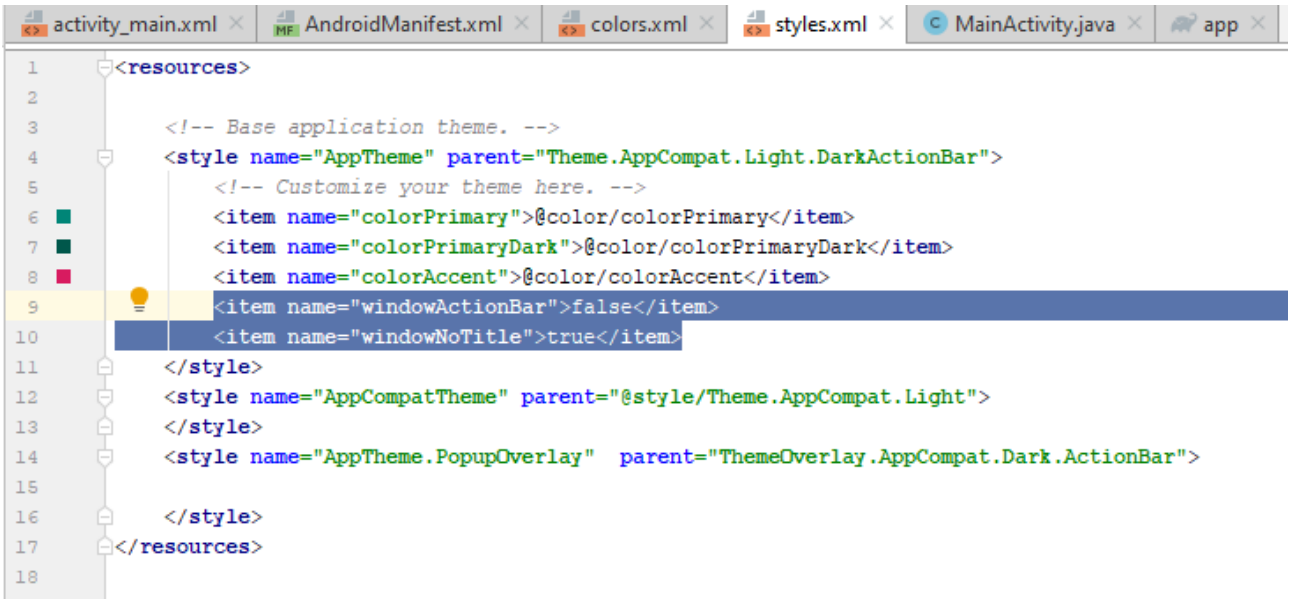




Juan Carlos Parralejo Martínez

1. Styles.xml.

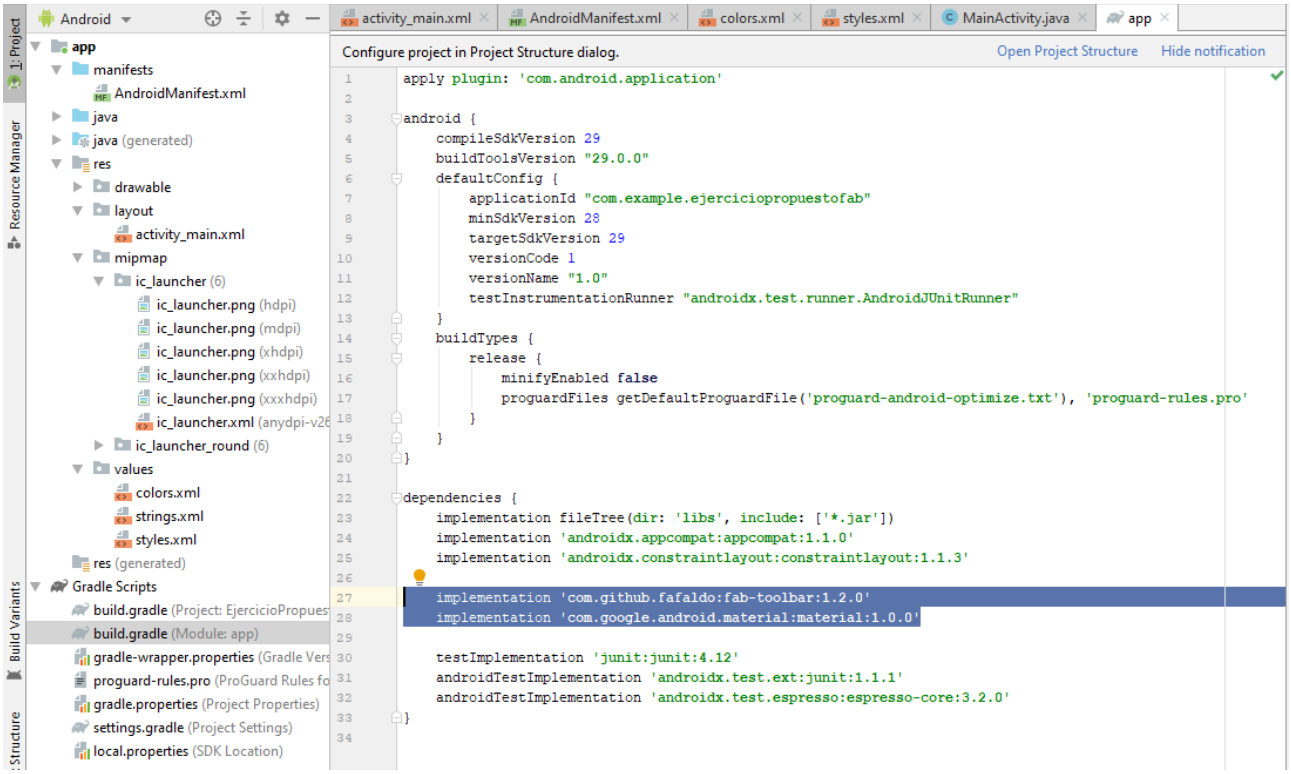
El primer paso que tenemos que hacer, al trabajar con AppBar es añadir las siguientes lineas al archivo styles.xml, indicando que no usaremos la actionBar ni el título.



Si no cambiamos esto, la actionBar entra en conflicto con la Appbar.

2. Archivo build.gradle

En el archivo build.gradle de la aplicación(no del proyecto) alñadimos las siguientes lineas para añadir las dependencias para usar FABToolbarLayout y material design. Se hace de esta forma en la version 3.5 de Android Studio, ya qu no lo sugiere en Project Structure – Dependencies.



3. Archivo activity_main.xml

Es necesario que el contenedor principal sea un CoordinatorLayout, dentro de este, en primer lugar crearemos la ToolBar como hicimos en el tema 3, como un elemento independiente.

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/ThemeOverlay.AppCompat.ActionBar">
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/Theme.AppCompat.Light"/>
```

El segundo componente es un FABToolbarLayout, el cual contiene el FloatingActionButton y el contenido de la ToolBar.

```
<com.github.fafaldo.fabtoolbar.widget.FABToolbarLayout
    android:id="@+id/fabtoolbar"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:containerId="@id/fabtoolbar_container"
    app:fabId="@id/fab"
    app:fabToolbarId="@id/fabtoolbar_toolbar"
    app:fadeInFraction="0.2"
    app:hideDuration="600"
    app:horizontalMargin="8dp"
    app:showDuration="600"
    app:verticalMargin="16dp">

    <!-- <include layout="@layout/activity_main"/> -->
```

Con app:containerId hacemos referencia a la ToolBar, con app:fabId hacemos referencia al FAB y con app:fabToolbarId hacemos referencia al contenido de la ToolBar.

Con la anotacion app:fadeInFraction indicamos el procentaje de desplazamiento en la animación y con app:showDuration, la duración de esta.

Dentro del FABToolbarLayout meteremos el FAB y el contenido de la ToolBar.

```
<RelativeLayout
    android:id="@+id/fabtoolbar_container"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/add"
        android:backgroundTint="@color/azul"
        app:borderWidth="0dp"
        app:fabSize="normal"/>
</RelativeLayout>
<LinearLayout
```

```
</RelativeLayout>
<LinearLayout
    android:id="@+id/fabtoolbar_toolbar"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:layout_alignParentBottom="true"
    android:orientation="horizontal">
    <ImageView
        android:id="@+id/uno"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:scaleType="centerInside"
        android:src="@drawable/account"
        />
```

4.Archivo MainActivity.java

En este creamos todos los Objetos del xml mendiante el método findViewById() y creamos los escuchadores de los botones implementando en la clase View.OnClickListener.

En el método onClick, mostraremos lo que hemos creado en en activity_main.xml con el método de FABToolbarLayout, show().

Tambien destacat el método setSupportActionBar que establece la barra de herramientas como la barra de app de la actividad.